

<Name-of-Software-Application>

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 7/16/2022 | Wallace, Eric | Software design document for The Gaming Room |

**Instructions**

Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

## [Executive Summary](#_sbfa50wo7nsh)

* Client Name: The Gaming Room
* Business Industry: Entertainment
* Description Of Game: Our client, The Gaming Room has hired our company to develop a cross-platform web-based game. The name of the game is “Draw It or Lose It” and consists of one or more teams competing for four rounds with each round last a minute. The premise behind the game is teams are given a puzzle in the form of an image and has a given amount of time to guess it, if they are unsuccessful then the remaining teams are given 15 seconds to guess the puzzle.

## [Design Constraints](#_2et92p0)

* One or more teams consisting of multiple people
* Cross-platform
* When selecting a team name, provide feedback to the team if selected team name is in use
* Generate a unique identifier for each game created so all game names are unique
* Only allow one game to exist and within that game only allow one puzzle to be present at one time.
* Timer system for tracking the amount of time each team has remaining to guess a puzzle.

## [System Architecture View](#_ilbxbyevv6b6)

Please note: There is nothing required here for these projects, but this section serves as a reminder that describing the system and subsystem architecture present in the application, including physical components or tiers, may be required for other projects. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should be provided.

## [Domain Model](#_8h2ehzxfam4o)

* **Entity –**Entity is an abstract class that has a inheritance relationship between Game, Team and Player.This means Entity is a base class and the Game, Team and Player classes inherit from Entity.
* **Teams –** Has a zero to many relationships with Player meaning there can zero to many Players on a single team. Teams also have a zero to many relationships with Games with means there can be multiple teams in one game.
* **GameServices –** It is also a zero to many relationships with Game which states there are multiple games in one GameService.
* **ProgramDriver –** ProgramDriver uses SingletonTester to perform evaluation on the cost to test for errors

| **Development Requirements** | | **Mac** | | **Linux** | | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Server Side** | **Characteristics:** Easy setup for both SSL and Non-SSL web servers  **Advantages:** Web server software is installed on the OS by default, implementing SSL is trivial and documentation is readily available. Web server software is open source and does not cost other than hardware needs  **Disadvantages:** OS does not technically have a server version so scaling it to large groups of users may not be possible.  MAC OS offers a relevant solution to host a web app, having the software installed on the OS by default allows for an easier implementation and setup. However, the lack of a server-based version of the OS may limit the number of simultaneous users. Server based deployment is straight forward, since the web server used is a very popular software deploying it usually means copying the server files to another server. | | **Characteristics:** Most commonly OS used for web hosting  **Advantages:** Has the largest community so documentation is easily accessed. All software needed is free and open source.  OS is the most secure out of the major 3.  **Disadvantages:** Configuration of the OS itself can be a daunting task for users not familiar with the OS. Setup is completed through the command line.  Linux is the mostly wide used OS for web servers. The web server software used in most implementation is open source and is free. Since Linux is offered in a server version, scaling it up to host thousands of users is not an issue for this OS. Deployment is easy as there is a plethora of applications available for deploying multiple web servers on the same machine can be easily implemented. One of the downsides to Linux is the server versions usually do not have a GUI so all software installation and setup would have to be done through the command line which means it does have a learning curve associated with it. However, the lack of a GUI is also one of the factors that makes Linux such a secure OS. Overall, the cost of using Linux to host web apps is relatively low and is only affected by the cost of the hardware. | | | **Characteristics:** Configuration is done so through a user interface  **Advantages:** Since it uses a GUI for configuration and setup can be more intuitive. Access to 3rd party tools and apps has more compatibility with Windows than with the other two Os’s.    **Disadvantages:** Tends to have more security vulnerabilities and more susceptible to DDOS attacks. Configuration of IIS is not as straight-forward given it has a GUI for setting up servers.  Windows is not as widely used as Linux when using it as a web server. The cost associated with using Windows as a web server can be quite expensive as the client would have to purchase the OS. Windows is also perfectly capable of hosting a web app that will be access by thousands of people simultaneously and deploying a web server is not too difficult to setup and replicates to other servers automatically once setup. Overall, even though setup would be through a GUI and server-based deployment is easy to setup most of the time, it is the cost of server software and hardware that eliminates Windows from being the chosen to host web apps. | **Characteristics:** Not widely heard of.  **Advantages:** There is no advantage to having a mobile device server and not event  **Disadvantages:** There is no true Server version |
| **Client Side** | | The development process required to support compatibility with all web browser platforms would be to use a front-end framework like React along with a CSS library like Tailwind. Utilizing HTML, CSS and JavaScript will make the app compatible on all modern web browsers without the need of writing code for each individual platform. Using a framework will help reduce the amount of time to complete the app which will decrease the cost of developing the app. Experienced developers will sometimes have custom libraries with components which allows them to implement those components into an app, which greatly reduces the amount of time required to develop an app because it reduces the amount of code that must be written. This also reduces the cost of developing an app. However, the expertise level dramatically increases whenever you make use of a framework and if a developer is not experienced in that framework, it can have the opposite effect on a project. | | | The same app which is developed for computers can be used with mobile apps. If an app is developed correctly the same server can be used to serve a web app. By developing a app that is responsive by using breakpoints the app can behave different on smaller screens than it does on larger screens. During the development of an app for compatibility on mobile devices it cannot be overstated the importance of a screen size. This is where using a CSS library/framework is priceless, most modern, and widely used CSS framework such as bootstrap, tailwind and materialui has all of that figured out and has it baked into the framework by default. This can help reduce the amount of time It takes to develop a web app and since the time to delivery has decreased so has the cost of developing the app. In regard to expertise, I would say the development of an app for mobile devices regardless if it is a web app or native app is going to be higher than that of a larger screen device. Using a framework decreases the level of expertise because writing the code to deal with the various screen size takes quite a bit of expertise which is eliminated using a framework but using a framework also increases the level of expertise because a developer must know the framework. | | |
| **Development Tools** | | The development of a web app can be done in a variety of ways. Most often HTML, CSS and JavaScript are the languages used to develop web apps.  For IDEs, one of the most popular IDEs is Visual Studio code which is available from all three OS’s. How familiar a developer is with an IDE can reduce the time it takes to develop an app. Using things such as snippets to auto-insert code can also help.  For the development of a web app the environment must be taken into consideration. One of the most common setups to develop a web app would be to use a run-time environment such as Nodejs to create a web server so developers can see how the code they are writing looks and feels. Setting up a development environment most often can be done from any OS.  Depending on the complexity and size of a web app will dictate if more than one development team is required. Sometimes there will be a team developing the front-end while another team develops the API/back-end. Using multiple teams to develop an entire app is very common and does reduce the amount of time required to complete an app.  Writing a web app will most likely be done in HTML, CSS and JavaScript and since most developers know the languages the technical impact it has on a development team is not that significant. Should the requirement change and a different language such as C# or Python with Flask, the impact could greatly increase especially if developers on the team are as experienced in that language. The impact an IDE has on a development team, I would say it is relative to how the IDE will be used. Most IDEs have features that set them apart from other IDEs. If members of a development team leverage the features of an IDE to develop code the impact could be quite substantially. Using an IDE that the team is familiar with then the team can focus on developing the app instead of trying to figure out how the software works. Depending on the IDE members of a development team uses will dictate if there is a licensing cost associated with it and if so, how much. Take for instance, VS Code, one reason why the software is so widely used is because it is free and has a lot of support for various extensions/plugins. At the end of the day, almost everything done during the development of an app will impact the development team, the cost, and the time to delivery which is usually related to the level of expertise required for development. | | | | | |

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## [Evaluation](#_2o15spng8stw)

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client’s requirements, and look at the situation holistically, as it all has to work together.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

**Recommendations**

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

1. **Operating Platform**: For the operating system or platform, the recommendation is to utilize Kubernetes Clusters on the Google Cloud Deploy product or solution. In a previous module the customer indicated they wanted a mechanism for tracking games, if that be the case they would also need an API server which is another service offered by Google, then using a database services the client could then implement the game tracking feature.
2. **Operating Systems Architectures**: The Cloud Deploy solution uses Linux servers with Kubernetes for deploying software, maintaining current version and update releases. This allows the client to be free of maintaining and supporting on-premises servers.
3. **Storage Management**: For storage, Cloud Deploy uses a simple Cloud Storage container which charges by the per GB used and is scalable to some arbitrary amount like 20 Petabytes. But since we are going to be deploying this app with the Cloud Deploy service the lowest amount of 1TB will do and they all use SSD’s and Fibre connections for communication.
4. **Memory Management**: Memory management is tricky when it comes to cloud services because most of them don’t even take into consideration the amount of RAM used on pricing, they usually charge per GB transmitted, number of seconds a resource was used for or the amount of runtime a server has. In our scenario and since we are only using this service for deployment and really nothing more a E2-Medium server is what Google recommends and it has 4GB of memory. The amount of RAM is not scalable so the introduction of resources monitoring might be helpful to determine if the server resources get low.
5. **Distributed Systems and Networks**: The Cloud Deploy solution is a Distributed System and Networks in a way since withing this one service is the storage, building, logging system, networking system all wrapped up together. Even though there are several different Google products and services being used in the process. Using GCP as your distributed system it allows you to scale depending on the size and need of your application while delivering 99.9% uptime with failover, so you never have downtimes.
6. **Security**: For security I would recommend adding Firebase to control authentication and CloudIAM for accessing resources in the Google Platform. Using Firebase and the APIs associated with it the security of the game will be very secure. Using the two different Google products two provide security means you are separating the two systems which means if someone were to ever break the security on the game all they would ever be able to access would be game data. With the Cloud Deploy solution you can enable push notifications through firebase which means you don’t have to necessarily have the client communicate with the backend servers, you can utilize communication through firebase which doesn’t open the API to the public but use Firebase to communicate with the GCP resources internally on the backend.